EXHIBIT 3

Modified Redacted Version of Dkt. 1112-20

HIGHLY CONFIDENTIAL - ATTORNEYS' EYES ONLY

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA – OAKLAND DIVISION

CHASOM BROWN, WILLIAM BYATT, JEREMY DAVIS, CHRISTOPHER CASTILLO, and MONIQUE TRUJILLO, individually and on behalf of all similarly situated,

Plaintiffs,

v.

GOOGLE LLC,

Defendant.

Case No. 4:20-cv-03664-YGR-SVK

REBUTTAL EXPERT REPORT OF KONSTANTINOS PSOUNIS, PH.D.

August 30, 2023

TABLE OF CONTENTS

I. BACKGROUND AND QUALIFICATIONS

1. I have been retained by Google to analyze and respond to certain opinions proffered by Plaintiffs' retained expert Mr. Jonathan Hochman in the June 20, 2023 Second Supplemental Report of Jonathan Hochman ("Hochman Second Supp. Rep.") submitted in the above-captioned litigation.

2. I am a Professor and Associate Chair of Electrical and Computer Engineering and Professor of Computer Science at the University of Southern California, where I teach courses on networked distributed systems, probability and information theory. I joined the University of Southern California in 2003, after completing my PhD at Stanford University as a Stanford Graduate Fellow. I have published more than 100 technical papers in the field of networked distributed systems, which have been cited tens of thousands of times. I have also been awarded numerous grants and significant funding from the government and industry leaders to advance these fields. As a result, I have been named an Institute of Electrical and Electronics Engineers (IEEE) Fellow, the highest grade of membership, and a Distinguished Member of the Association of Computing Machinery (ACM) for my contributions to the theory and practice of networked, distributed systems. Attached hereto as Exhibit A is a true and correct copy of my curriculum vitae.

3. My professional career has spanned more than 20 years. As set forth in Exhibit A, I have extensive experience in the field of networked distributed systems, including the Internet and the world wide web, content-delivery networks, data centers and cloud computing, and wireless mobile networking systems. Throughout my career, I have analyzed, designed, and developed efficient, privacy-preserving networked distributed systems for the Internet and the Web, content-delivery networks, data centers and cloud systems, and wireless mobile networking systems. As such, I have acquired expertise in the analysis and development of those systems. I

have also been the faculty in charge of the entire networking curriculum at the Electrical and Computer Engineering department at USC for more than a decade and teach networking classes as well as probability theory classes which cover entropy, statistics, and other related concepts to graduate students yearly. In my analysis of networked distributed systems and my associated technical publications I regularly use probabilistic and statistical approaches.

II. EXECUTIVE SUMMARY OF OPINIONS

4. **Opinion 1 (§ III):** The [redacted] bit does not support Mr. Hochman's opinions stated in his April 15, 2022 Opening Report ("Hochman Opening Report"), June 7, 2022 Supplemental and Rebuttal Report ("Hochman First Supp. Report"), or Hochman Second Supplemental Report because (i) the [redacted] bit does not resolve any of the flaws associated with Mr. Hochman's proposed methodology for identifying users in private browsing modes that I have previously identified; (ii) Mr. Hochman's characterization of the [redacted] bit as a "side channel attack" is misleading; (iii) Mr. Hochman's assertions regarding Google's purported failure to comply with World Wide Web Consortium ("W3C") Guidelines[1] regarding private browsing modes are incorrect; and (iv) the [redacted] bit has not worked since July 2019 for Chrome, September 2019 for Safari, and January 2020 for Edge, and it was developed for [redacted] purposes.

5. **Opinion 2 (§ IV):** The [redacted] bit does not change any of the opinions stated in my June 7, 2022 Rebuttal Report ("Rebuttal Report"), including my opinions regarding (i) Mr. Hochman's incorrect claim that users can be readily identified with the data at issue; (ii) Mr. Hochman's incorrect assertions regarding private browsing "profiles," Google's server-side processes, and data joinability; (iii) Mr. Hochman's misleading and unfounded claims regarding

[1] "W3C TAG Observations on Private Browsing Modes," World Wide Web Consortium, July 5, 2019, https://www.w3.org/2001/tag/doc/private-browsing-modes/#accessible-private-normal.

Google's fingerprinting policies and practices; (iv) Mr. Hochman's unworkable proposals for identifying class members; (v) Mr. Hochman's inaccurate claims regarding the accuracy of Incognito and private browsing detection methods; and (vi) Mr. Hochman's failure to account for shared devices.

6. **Opinion 3 (§ V):** The [redacted] logs containing the maybe_chrome_incognito bit that Mr. Hochman cites do not support the opinions stated in the Hochman Opening Report, Hochman First Supplemental Report, or the Second Supplemental Report because (i) the existence of this bit in additional logs does not resolve any of the flaws associated with Mr. Hochman's proposed methods for identifying class members that I identified in my Rebuttal Report; (ii) the "[redacted]" log does not allow "joining" or "linking" of data from signed-out private browsing sessions with records associated with a user's Google account; and (iii) adding records to the same log and sorting them by time stamp does not "join" or "link" signed-out private browsing mode data with signed-in browsing data.

7. **Opinion 4 (§ VI):** The [redacted] logs containing the maybe_chrome_incognito bit that Mr. Hochman cites do not change any of the opinions stated in my Rebuttal Report, including my opinions regarding (i) Mr. Hochman's incorrect claim that users can be readily identified with the data at issue; (ii) Mr. Hochman's incorrect assertions regarding private browsing "profiles," Google's server-side processes, and data joinability; (iii) Mr. Hochman's misleading and unfounded claims regarding Google's fingerprinting policies and practices; (iv) Mr. Hochman's unworkable proposals for identifying class members; (v) Mr. Hochman's inaccurate claims regarding the accuracy of Incognito and private browsing detection methods; and (vi) Mr. Hochman's failure to account for shared devices.

III. [redacted] DOES NOT SUPPORT MR. HOCHMAN'S OPINIONS

8. I have reviewed the December 21, 2022 Declaration of Borbála Katalin Benkő ("Benkő Declaration") and a Google News Initiative blog post authored by Barb Palser cited by Mr. Hochman ("GNI Blog Post").[2] Based on those materials, my decades of experience in this field, and the sources cited below, I have reached the following conclusions.

A. [redacted] Does Not Substantiate Mr. Hochman's Prior Opinions

9. Mr. Hochman states that "Google's development and use of the [redacted] private browsing detection bit further substantiates the following opinions that I have already offered: Opening Report Opinions 1, 2, 4, 5, 6, 7, 8, 17, 18, 19, 20, 22, 24, 25, 26, 27, 29, and 31, and Rebuttal and Supplemental Report Opinions: 1, 3, 5, and 6. These opinions address, among other topics, Google's interception, storage, and use of private browsing data, Google's ability to identify private browsing traffic and class members, and Google's ability to delete the private browsing data it has already collected." Hochman Second Supp. Rep. ¶ 22. For the reasons discussed below, the [redacted] bit does not substantiate these opinions because (i) each of Mr. Hochman's proposed methods for identifying users in private browsing mode relies on his incorrect claim that a combination of an IP address and user agent value (or a number of pseudonymous identifiers) can be used to match records associated with a given user's signed-in non-private browsing with records of the same user's signed-out private browsing; and (ii) the existence of the [redacted] bit does not change this fundamental flaw in any way.

10. As I explained in my June 7, 2022 Rebuttal Report ("Rebuttal Report"), Mr. Hochman's proposed method for identifying users in private browsing mode will not work because the combination of IP address and user agent values cannot be used to reliably match

[2] *See* Hochman Second Supp. Rep. ¶ 17 n.2 (citing Barb Palser, Protecting Private Browsing in Chrome, Google News Initiative (originally posted on July 19, 2019; updated on July 21, 2020), https://blog.google/outreach-initiatives/google-news-initiative/protecting-private-browsingchrome/).

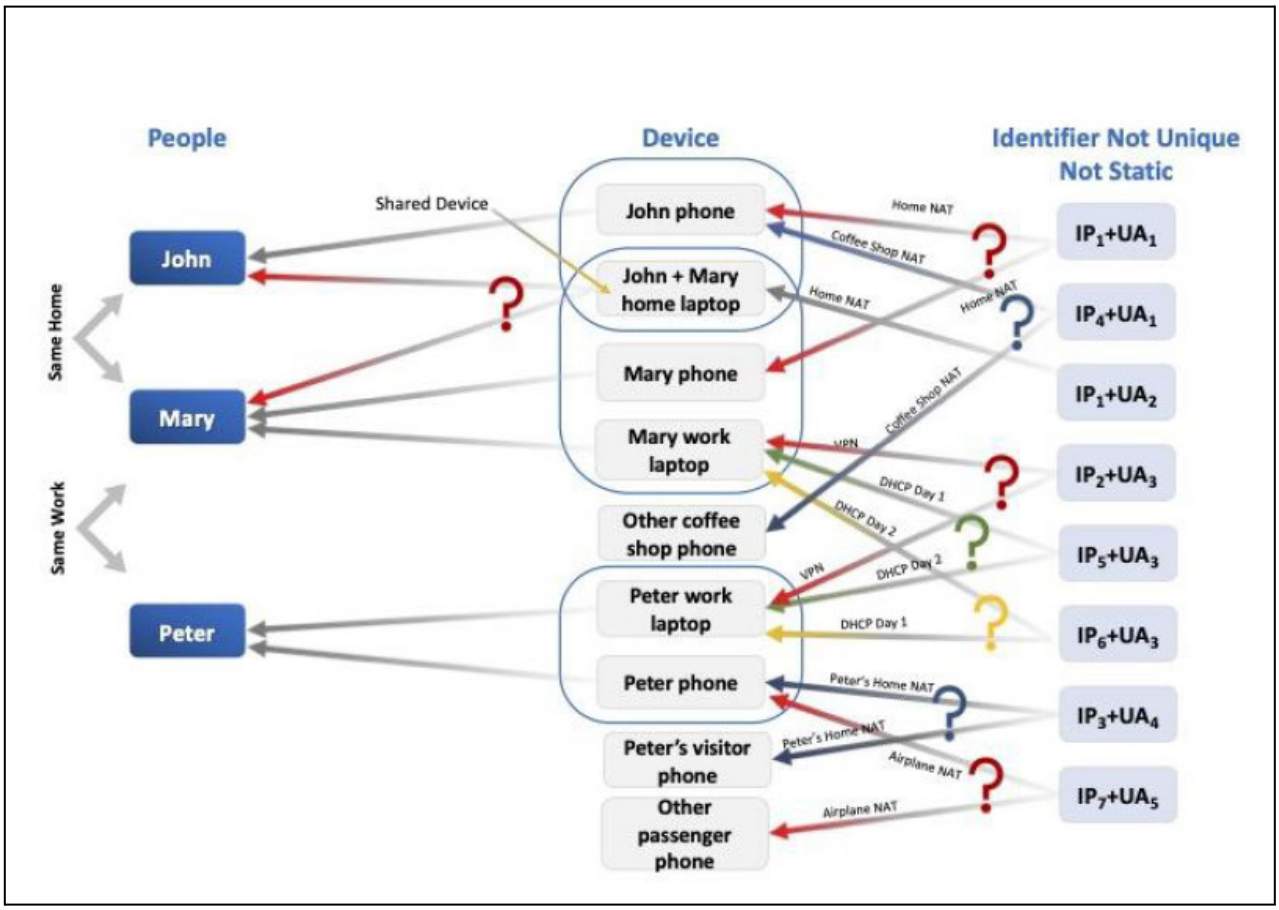
records of signed-out private browsing mode activity. *See* Rebuttal Report §§ III.A, III.G, III.I, III.J, IV.B, V.E–F.

11. The same is true of Mr. Hochman's proposed use of pseudonymous identifiers (*e.g.*, PPIDs, PPID-mapped Biscottis, User IDs, and Biscotti IDs). *See* Rebuttal Report §§ III.A, III.G, III.I, III.J, IV.B, V.E–F. These pseudonymous identifiers cannot be used to identify users of private browsing modes for the reasons I explained in my Rebuttal Report.[3] Adding the [redacted] bit to logs containing these identifiers does not change anything about these identifiers, and thus Mr. Hochman's proposed methodology for matching data associated with a user's Google account and signed-out private browsing mode data will not work.

12. The existence of the [redacted] bit would not do anything to resolve this flaw in Mr. Hochman's proposed methodology because it does not change anything about the IP address and user agent values. For the reasons discussed in my Rebuttal Report (§§ III.A, III.G, III.I, III.J., IV.B, V.E–F) an IP address and user agent cannot be used to reliably link signed-out private browsing mode records with signed-in non-private browsing mode records because these two values are not reliably mapped to individual users. This problem is illustrated by the following demonstrative, where the use of NAT, VPN, DHCP and shared devices makes it impossible for the combination of an IP address and user agent to be reliably mapped to an individual:

[3] *See id.* (explaining that Mr. Hochman's proposed use of pseudonymous identifiers cannot reliably identify class members, including because (i) only a subset of users will even be assigned a pseudonymous identifier (i.e., if they opt-in to Chrome metrics for UMA ID or if they visit websites that use PPID and/or User ID and the user signs-in to their account on those websites); (ii) UMA data is designed for aggregate analysis and is not joined with authenticated identifiers; and (iii) PPID and/or User ID are assigned by websites, purposely processed before they are sent to Google to ensure they do not contain personal identifying information, and will have the same value for users who share an account for a given website).

Figure 1 - Lack of Reliable IP Address and UA Mapping to Individual Users



13. Adding the [redacted] bit to the IP address and user agent combination depicted in Figure 1 above will not do anything to resolve the lack of a reliable mapping between IP address, user agent, and individual users. Nor would it do anything to resolve the issues posed by shared devices that I explained in detail in my Rebuttal Report. *See* Rebuttal Report §§ III.J, V.E–F. For this reason, Mr. Hochman's assertion that the existence of the [redacted] bit "further substantiates" his opinions regarding "Google's ability to identify . . . class members" is incorrect. Hochman Second Supp. Rep. ¶ 2.

14. [redacted] also does not support Mr. Hochman's opinions regarding Google's ability to reliably identify private browsing traffic. *See* Hochman Second Supp. Rep. ¶ 22, 27, 30. As Ms. Benkő explained, "[redacted]

[redacted]." Benkő Decl. ¶ 5. The [redacted]" *Id.* It is not surprising that this method would lead to false positives, as there are a number of ways that a user can alter the ability of a browser to call these APIs and/or affect how they respond without browsing in private mode via user preferences and settings. *See, e.g.*, https://stackoverflow.com/questions/74728235/disable-local-storage-but-not-cookies-in-edge (discussing reproducing a "hardened" environment where local storage is blocked); https://stackoverflow.com/questions/6600754/how-can-i-browse-with-localstorage-disabled (discussing how to browse the web with local storage and cookies disabled); https://blog.tomayac.com/2022/08/30/things-not-available-when-someone-blocks-all-cookies/#:~:text=Turns%20out%2C%20with%20all%20cookies,localStorage (explaining how blocking cookies using Chrome's settings will disable local storage). If, for example, a user were to change his or her browser settings to disable local storage and/or disable these APIs (or install a plugin or extension to do so), I would expect that the [redacted] field would be set to "true" even if the user is not using private browsing mode.

15. In light of the false positives and false negatives Ms. Benkő identified, *see* Benkő Decl. ¶ 5, [redacted] also does not support Mr. Hochman's opinions regarding Google's purported ability to "delete the private browsing data it has already collected." Hochman Second Supp. Rep. ¶ 22, 31, 49. [redacted] can return a false positive and a false negative value and its accuracy cannot be independently verified for a given record, and Mr. Hochman does not propose a method for addressing these false positives and false negatives. Unless the false negatives are identified and addressed, Google cannot delete all private browsing data it has

already collected. And, unless the false positives are identified and addressed, Google cannot, for example, simply delete all logs that contain a value of "true" for the [redacted] bit without deleting these false positives (*i.e.*, deleting data that was collected from a regular browsing session).

16. Note that as I have explained in my Rebuttal Report, maybe_chrome_incognito also does not support Mr. Hochman's opinions regarding Google's purported ability to "delete the private browsing data it has already collected." Like [redacted] maybe_chrome_incognito also suffers from both false positives and false negatives. *See* Rebuttal Report ¶ 143; Berntson June 16, 2021 Dep. Tr. 374:4–21. Hence, it cannot be used to delete all private browsing data, and, if used to delete some private browsing data it will also delete data that was collected from a regular browsing session.

17. Last, Mr. Hochman states that his data analysis for the is_chrome_incognito and maybe_chrome_incognito bits "indicate that they detect Incognito traffic extremely accurately." Hochman Rebuttal and Supplemental Report ¶ 28. As a threshold matter, to appropriately analyze the accuracy of the aforementioned bits in detecting Incognito traffic using data analysis one must use an appropriate data sample, referred to in statistics as a representative sample. *See, e.g.*, Mendenhall, W., and Sincich, T., *Statistics for Engineering and the Sciences*, 6th Ed., CRC Press (2016); William G. Cochran, *Sampling Techniques*, 3d Ed., John Wiley & Sons (1977); Hogg, R. V., Tanis, E. A., and Zimmerman, D., *Probability and Statistical Inference*, 10th Ed., Prentice Hall (2019). Mr. Hochman is basing his analysis on a tiny sample consisting of data from only six users (the five plaintiffs and one expert), which is not a representative sample of the general population of hundreds of millions of users. In addition, this non-representative sample may also be biased. As a result, one cannot use this data to estimate the accuracy of the aforementioned bits in detecting Incognito traffic in the general population, and, Mr. Hochman's accuracy number is

invalid. *See id*. Second, not surprisingly, the invalid accuracy number reported by Mr. Hochman is quite far from the false positives rates calculated using large representative samples by Google engineers, for example, the reported 1.5-2% inaccuracy because of just one of the many ways that a false positive may occur. *See* Berntson June 16, 2021 Tr. 373:22-376:11; GOOG-BRWN-00845673, at -674 ("There is one condition in which the header also isn't sent, which we cannot exclude: If the browser hasn't received any experiment config yet, it will not send the header. This accounts for roughly 1.5-2% of browsing."). Third, the overall inaccuracy of the aforementioned bits in detecting Incognito traffic could be much larger than this number as all false positive scenarios add up, but, even if it is 1.5-2%, this would result in tens of millions of mischaracterized data in view of the scale of the collected data at issue.

B. Mr. Hochman's Opinion Regarding A Purported Side Channel Attack Is Misleading

18. Mr. Hochman also asserts that [redacted] "qualifies as a 'side channel attack.'" Hochman Second Supp. Rep. ¶ 29 (citing "Side-Channel Attack," Computer Security Resource Center, https://csrc.nist.gov/glossary/term/side_channel_attack (last accessed June 15, 2023). According to Mr. Hochman, "side channel attacks" refer to any "computation that utilize[s] unintentionally leaked data from computer software or systems." Hochman Second Supp. Rep. ¶ 29 n.4. Based on my experience and training, Mr. Hochman's labeling of the [redacted] as an "attack" is misleading. Side channel analysis can be used to stage an "attack" by a hacker or other bad actor in order to exploit a vulnerability ("for bad"), but side channel analysis can also be used "for good." For example, the Ericsson article cited by Mr. Hochman (*see id.* (citing Ericsson.com, https://www.ericsson.com/en/blog/2023/4/side-channel-analysis (last accessed June 18, 2023)), makes the same distinction, as it explains that "side channels" may be used "for good" because "in some scenarios it is valuable to be able to externally determine the health and state of the device by observing the processes being performed in it." *Id.* As Ms. Benkő explained, Google

[redacted]

[redacted]." Benkő Decl. ¶ 4. In my opinion, [redacted]

[redacted]

[redacted]

19. I note that Mr. Hochman does not state that this categorization supports or substantiates any of his prior opinions.

C. Mr. Hochman's Opinion Regarding Google's Compliance With The W3C Guidelines Cited In My Rebuttal Report Is Incorrect

20. Mr. Hochman also contends that "Google sends itself signals (including the X-Client-Data Header and the [redacted] API) and uses these signals to actively detect private browsing data and tag that data as 'private,' in violation of the W3C guideline on which its own expert relied." Hochman Second Supp. Rep. ¶ 33. Mr. Hochman further opines that "Dr. Psounis either ignored these Google practices for [the] purposes of his report, or Google withheld this information as well." I disagree with Mr. Hochman's assertion because (1) neither the absence of the X-Client-Data header nor the [redacted] bit are reliable means for detecting Incognito or other private browsing; (2) Google's use of the absence of the X-Client-Data header and the [redacted] to approximate the use of private browsing modes do not violate the W3C guidelines; and (3) after the W3C Guideline was published, Google quickly moved to modify the Chrome browser to align with these Guidelines.

21. As to the X-Client-Data header, I explained in my Rebuttal Report (§§ III.H that the absence of the X-Client-Data header in a given record does not accurately identify private browsing data in light of the multiple instances where the X-Client-Data header is not sent when a user is not using Incognito mode ("false positives"). Because it (a) is not a signal that "Google

sends" and (b) does not reliably detect the usage of Incognito mode, the absence of the X-Client-Data header does not constitute a "signal" identifying the use of Incognito mode.

22. As to the [redacted] bit, this is not a "signal" sent by the browser at all—[redacted]. And as I explained above, like the absence of the X-Client Data Header, this bit does not reliably detect Incognito usage because there are [redacted]

23. The W3C Guideline cited in my Rebuttal Report (https://www.w3.org/2001/tag/doc/private-browsing-modes/) is aimed at preventing websites from "degrad[ing] browsing experience (for example, not displaying content) when they detect the users in private browsing modes." It thus encourages browser developers to work toward making private browsing undetectable by websites in order to prevent websites from changing the user experience for a user in private browsing mode. The maybe_chrome_incognito bit was developed to provide internal aggregate metrics on approximate Incognito usage. *See, e.g.*, Rebuttal Report ¶ 143 (citing, *inter alia*, GOOG-CABR-04470006, at -009 ("For Chrome, we have the x-client-data header in addition as a signal. But again, this will be heuristics-based, and can never [be] 100% accurate. The goal is to keep an eye on an envelop[e] of such traffic."). And the [redacted]. Benkő Decl. ¶ 5. In my opinion, these approximation efforts do not degrade the user experience and are not what the W3C guidelines were designed to prevent.

24. Additionally, I note that the W3C Guidelines were published on July 4, 2019. Google published the GNI Blog Post two weeks after publication of the W3C Guideline, on July 18, 2019. The GNI Blog Post announced that Google modified "the behavior of the FileSystem API" with the release of Chrome 76 "to remedy this method of Incognito Mode detection." This indicates that after the W3C Guideline was published, Google quickly moved to modify the Chrome browser to align with these Guidelines. With respect to the non-Chrome browsers at issue in this case, Google would not control the design of these browsers, and thus it cannot ensure that Apple and Microsoft comply with industry guidelines.

D. [redacted], And It Has Not Worked Since At Least July 2019 for Chrome, September 2019 for Safari, and January 2020 for Edge

25. Mr. Hochman states that "[i]t is also possible that the [redacted] bit still functioned well into 2020" and asserts that [redacted] "presumably remained operational . . . as of at least July 2020, if not longer." Hochman Second Supp. Rep. ¶ 17. As discussed further below, I disagree with Mr. Hochman's assertion because the Incognito logic was set in 2016 and was not changed, but the behavior of the APIs on which it relied were modified (thereby breaking the heuristic) by July 2019 for Chrome, September 2019 for Safari, and January 2020 for Edge.

26. The Benkő Declaration states that "Starting on [redacted]'" Benkő Declaration ¶ 3. Ms. Benkő explains that [redacted]

[redacted] *Id.* ¶ 4.

27. The Benkő Declaration further explains that [redacted] ." *Id.* ¶ 5. Ms. Benkő also notes that she is "[redacted] ." *Id.*

28. [redacted] *Id.* ¶ 5.

29. [redacted] (Benkő Decl. ¶ 5):

- [redacted]

30. The GNI Blog Post explained in July 2019 that "[t]oday, some sites use an unintended loophole to detect when people are browsing in Incognito Mode. Chrome's FileSystem

[4] I understand that Plaintiffs voluntarily excluded users of the Firefox browser from their proposed Class II on June 21, 2022. *See* 608-3 (Plaintiffs' Motion for Class Certification) at 1.

API is disabled in Incognito Mode to avoid leaving traces of activity on someone's device. Sites can check for the availability of the FileSystem API and, if they receive an error message, determine that a private session is occurring and give the user a different experience." It further explained that, "[w]ith the release of Chrome 76 scheduled for July 30, the behavior of the FileSystem API will be modified to remedy this method of Incognito Mode detection." As Ms. Benkő explained in her declaration, Chrome's change to "[redacted]. Benkő Declaration ¶ 6.

31. In the July 2019 GNI Blog Post, Google contemplated that there would be further changes to Chrome as additional methods of identifying Incognito mode were discovered, explaining that "Chrome will likewise work to remedy any other current or future means of Incognito Mode detection." Shortly after Chrome 76 was released, websites began finding other methods to exploit the FileSystem API to detect Incognito mode, which was publicly reported on.[5] Unsurprisingly, in July 2020, the GNI Blog Post was updated to explain that Chrome was gradually rolling out another fix to "address a loophole that could be used by websites to detect Chrome Incognito Mode sessions," which "was first announced in January [2020]." The update further explained that yet "[a]nother change announced in January [2020]" was "rolled out with Chrome 80 in February." However, [redacted]. Benkő Declaration ¶ 6 ("[redacted]

[5] *See, e.g.*, *Websites Still detecting Chrome Incognito Mode despite loophole fixed by Google* (techdows.com) (August 9, 2019) ("If you realized Google just fixed the loophole in the Chrome that prevents sites from detecting incognito mode, sites got clever and started using other methods to detect Private mode in Chrome 76. Chromium team itself aware of the fact that it is still possible to detect Incognito using File System API with Quota and timing attacks."); *see also Here's Why Some Sites Can Still Tell You're In Incognito Mode* (gizmodo.com) (August 12, 2019) (explaining "[t]he fix in Chrome 76 made the FileSystem API available even when in Incognito Mode, meaning sites would no longer be able to use its absence as a means of sussing out what mode someone was browsing in," but noting the New York Times had already "figured out another loophole").

[redacted]

[redacted]").

32. I would also expect the [redacted] bit to have ceased functioning for the Safari browser when Safari's private mode's use of the localStorage API for the Safari browser was modified in a way that rendered private browsing mode detection using this API ineffective at least by the September 20, 2019 release of Safari 13. *See* https://stackoverflow.com/questions/42182648/detect-safari-private-browsing-in-applescript ("[T]his no longer works with Safari 13. (And I believe some earlier versions as well.)"); https://osxdaily.com/2019/09/19/safari-13-released-for-mac/ (announcing September 19, 2019 Safari 13 release date).

33. I would also expect the [redacted] bit to have ceased functioning for the Microsoft Edge browser by at least January 15, 2020, when Microsoft's Edge browser was updated to use the open-source Chromium engine. *See* https://www.browserstack.com/blog/chromium-based-edge/ ("Microsoft rolled out the latest version of its Edge browser, Edge 79, on January 15, 2020. This update, codenamed 'Project Anaheim', is a landmark shift for Microsoft, from the EdgeHTML engine to the Chromium engine.").

34. In light of these changes, Mr. Hochman's assertion that "[redacted] presumably remained operational in Safari, Firefox, Edge, and Edge Legacy as of at least July 2020, if not longer" is plainly incorrect because each of the APIs on which [redacted] relies were changed no later than July 2019 for Chrome, September 2019 for Safari, and January 2020 for Edge. Hochman Second Supp. Rep. ¶ 17. This is consistent with Ms. Benkő's explanation that the [redacted]" at least as of the date of her December 21, 2022 declaration.

35. Additionally, Mr. Hochman's assertions regarding Google's ability to delete private browsing data using the [redacted] bit, Hochman Second Supp. Rep. ¶ 22, 31, 49, ignores the fact that (i) with limited exceptions, Google employs an [redacted] upper-bound retention period for unauthenticated data (including data from signed-out private browsing), GOOG-BRWN-00028990 at -991; and (ii) the retention timeframe for data used for security/fraud/abuse purposes is less than [redacted], *id.* at -993. As explained above, Mr. Hochman's assertion that the [redacted] bit functioned until July 2020 is incorrect. However, even if the bit still worked until July 2020, I would expect data that is subject to these retention periods from July 2020 to have already been deleted no later than July 2023. [redacted] could not be used to delete data from private browsing modes for this additional reason.

IV. [redacted] DOES NOT CHANGE ANY OF MY PRIOR OPINIONS

36. Mr. Hochman states that "Google's development and use of the [redacted] private browsing detection bit further substantiates the following opinions that I have already offered: Opening Report Opinions 1, 2, 4, 5, 6, 7, 8, 17, 18, 19, 20, 22, 24, 25, 26, 27, 29, and 31, and Rebuttal and Supplemental Report Opinions: 1, 3, 5, and 6." Hochman Second Supp. Rep. ¶ 22. In my Rebuttal Report, I rebutted Mr. Hochman's opinions in sections III.A, III.C, III.F, III.G, III.H, III.J, IV.B, and V.E–F. The addition of the [redacted] boolean field to records stored in logs does not change any of those opinions.

37. **Rebuttal Report Section III.A (Opinion 1): "Mr. Hochman's Opinion That Users Can Readily Be Identified From The Data At Issue (# 18) Is Incorrect."** The [redacted] bit does not change this opinion because the data-at-issue is orphaned and unidentified: (a) any unauthenticated data in these logs is still keyed to an unauthenticated identifier (or no identifier) for a signed-out user that is unique to the private browsing session; and

(b) this bit does not change the operation of the cookie jar and server-side processes for users in private browsing modes described in paragraphs 37 through 58 of my Report.

38. **Rebuttal Report Section III.C (Opinion 3): "Mr. Hochman's Opinions On 'Private Browsing Profiles,' Server-Side Processes, And Data Joinability (# 10, 18, 19, 20) Are Inaccurate."** The [redacted] bit also does not change this opinion because (i) it does not show that Google maintains "cradle-to-grave" profiles of users that join signed-out private browsing mode activity with a Google account; and (ii) it does not change Google's server-side processes designed to prevent the joining of authenticated and unauthenticated data.

39. **Rebuttal Report Section III.F (Opinion 6) "Mr. Hochman's Assertions On Fingerprinting Are Misleading And Unfounded."** The [redacted] bit does not change this opinion because it does not show that Google engages in fingerprinting to identify users in private browsing mode or otherwise undermine any of my opinions regarding the technical and policy constraints that Google implements to prevent the use of fingerprinting to re-identify users.

40. **Rebuttal Report Section III.G, III.I (Opinions 7 and 9): "Mr. Hochman's Proposal to Identify Class I (Chrome Class) Is Unreasonable And Unreliable," and "Mr. Hochman's Proposal to Identify Class II (Non-Chrome Class) Is Unreasonable And Unreliable."** The [redacted] bit does not change this opinion because the same issues described in my Report will also affect any attempt to identify class members by applying the same fingerprinting methodology to log records where the [redacted] bit is set to "true." *Id.* Mr. Hochman's proposed IP + UA fingerprinting method would still rely on combinations of IPv4/IPv6 addresses and user agents (or other unworkable identifiers) that are "not sufficiently unique to identify class members because there are many situations where more than one user will have an identical IP address and user agent," and the [redacted] bit does not resolve this fundamental flaw because it is inherent to the proposed use of a combination of an IP address and

user agent as a join key to identify class members. *Id.* ¶ 111; *see also* V.E–F (appendices rebutting Mr. Hochman's claims regarding entropy and the ability to identify users via Mr. Hochman's proposed fingerprinting methodology).

41. **Report Section III.H (Opinion 8): "Mr. Hochman's Opinion That The 'maybe_chrome_incognito' Bit Reliably Detects Incognito Traffic (# 23) Is Incorrect."** As I explained in my Rebuttal Report, the absence of the X-Client-Data header does not provide a reliable method for accurately identifying the use of Incognito mode because there are instances where the X-Client-Data header is not sent, but the user is not browsing in Incognito mode. As Ms. Benkő noted in her declaration, there are also instances where the [redacted] field will return false positives and false negatives. *See* Benkő Decl. ¶ 5 ("[redacted]"). Because of these false positives and false negatives, the inability to verify individual values, and the reality that it no longer functions for any browser at issue, the [redacted] bit does not change my prior opinions regarding Google's ability to reliably detect Incognito traffic.

42. **Report Section III.J (Opinion 10): "Mr. Hochman's Proposed Methods For Identifying Class Members (# 22) Do Not—And Cannot—Account For Shared Devices Or Accounts."** The [redacted] field does not change this opinion because the same issues described in my Report will also affect any attempt to identify class members by applying the same fingerprinting methodology to logs that contain the [redacted] bit. *Id.* Specifically, Mr. Hochman's proposed methodology still does not account for shared devices, which renders his methodology unreliable in light of widespread device-sharing and resulting collisions of IP

address and user agent combinations. *See Id.* ¶¶ 162-180; *see also* §§V.E–F (appendices rebutting Mr. Hochman's claims regarding entropy and the ability to identify users via Mr. Hochman's proposed fingerprinting methodology).

V. THE [redacted] ADDITIONAL LOGS DO NOT SUPPORT MR. HOCHMAN'S PRIOR OPINIONS

43. Mr. Hochman states that "Google's disclosure of . . . [redacted] additional logs with Incognito detection bits further substantiates the following opinions, which I have already offered: Opening Report Opinions 1, 2, 4, 7, 8, 11, 12, 13, 14, 17, 18, 19, 20, 22, 23, 24, 25, 29, 30, 31; and Rebuttal and Supplemental Opinions 1, 2, 3, 5, and 6. These opinions focus on Google's interception, storage, and use of private browsing data, as well as Google's ability to delete the data it has already collected and Google's ability to destroy algorithms developed through that data." Hochman Second Supp. Rep. ¶ 42.

44. For the reasons discussed below, the existence of Incognito detection bits in these [redacted] additional logs does not substantiate Mr. Hochman's opinions. Each of Mr. Hochman's proposed methods for identifying users in private browsing mode relies on his incorrect claim that a combination of an IP address and user agent value (or a number of pseudonymous identifiers) can be used to match records associated with a given user's signed-in non-private browsing with records of the same user's signed-out private browsing; however, the existence of the Incognito detection bits in additional logs does not change this fundamental flaw in any way. In other words, because Hochman's IP + UA fingerprinting method for identifying *users* simply does not work, the fact that certain logs contain incognito detection bits that may or may not indicate which *mode* a given user was in at the time is irrelevant. For the same reasons, these [redacted] logs also do not substantiate Mr. Hochman's opinions regarding Google's purported ability to delete data collected from users in private browsing modes or any programs or algorithms that have ever used such data.

A. The "[redacted]" Log Does Not Allow "Joining" or "Linking" Of Signed-Out Private Browsing Mode Records With Records From Signed-In Non-Private Browsing

45. Mr. Hochman claims that the "[redacted]" log "raises the question of why Google calls the log a '[redacted] Log' but claims there is no joining," and he asserts that the existence of this log "undermines any claim that Google employs best-in-class safeguards to prevent data joining." Hochman Second Supp. Rep. ¶¶ 43-44. Mr. Hochman further states that the "structure of this '[redacted] log' also reinforces [his] opinions about how private browsing is identifying" because the sorting of records by time "may appear in sequential order, linking the signed-out private browsing data with the user's GAIA ID." Id. ¶ 45. As explained further below, I disagree with Mr. Hochman's assertions because (i) the source code for this log does not permit "joining" or "linking" of any signed-in and signed-out records as those terms are commonly understood in technical literature and in the field; and (ii) adding records to a log does not make it any easier to join such records.

46. Based on my experience and training, "joining" of log data refers to associating separate log records with a shared key, such as an identifier. For the logs in question, authenticated and unauthenticated data would be considered "joined" if a log shows that a shared key (or any common data point) was used to associate or combine unauthenticated private browsing data at issue with an individual's Google account or with their signed-out regular mode data. This conforms to well-established definitions from a long line of technical literature defining "joining" or "linking" of datasets as merging records from multiple datasets into combined records (*i.e.*, records that contain information from more than one "input" dataset) via the use of a common join

key, and it aligns with the definition employed by researchers in this field for decades.[6] Common database programming languages like Python and SQL define "join" and "joining" in the same way.

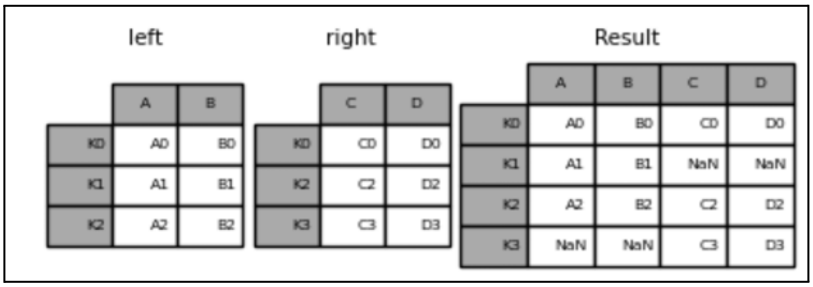
47. Python is one of the most widely used programming languages for data analysis today. In Pandas, which is a popular open source Python library for data analysis initially released in 2008,[7] the "join" function is used to combine records from two or more database-style "DataFrames" based on a common key between them, whereas the default "concat" function is used to concatenate records from two or more database-style "DataFrames" by merely listing them

[6] *See, e.g.*, Rob Hall and Stephen E. Fienberg, "Privacy-Preserving Record Linkage," 2010 Int'l Conf. on Priv. in Stat. Databases, https://www.cs.cmu.edu/~rjhall/linkage_survey_final.pdf at 1 (Sept. 2010) ("Record linkage is an historically important statistical problem arising when data about some population of individuals, is spread over several files. Most of the literature focuses on the two file setting. *The record linkage goal is to determine whether a record from one file corresponds to a record of a second file, in the sense that the records describe the same individual*." (emphasis added)); Ahmed K. Elmagarmid, "Duplicate Record Detection: A Survey," IEEE Transactions on Knowledge and Data Engineering, https://www.cs.purdue.edu/homes/ake/pub/TKDE-0240-0605-1.pdf, 19:1 at 1 (Jan. 2007) ("[T]he construction of a comprehensive view of . . . data [from multiple data sets] consists of linking—in relational terms, joining—two or more tables *on their key fields*." (emphasis added)); Ivan P. Fellegi and Alan B. Sunter, "A Theory for Record Linkage," Journal of Amer. Statistical Assoc. 64:328, https://courses.cs.washington.edu/courses/cse590q/04au/papers/Felligi69.pdf at 51 (Dec. 1969) ("The necessity for comparing the records contained in a file L_A with those in a file L_B in an effort to determine which pairs of records relate to the same population unit is one which arises in many contexts, most of which can be categorized as either (a) the construction or maintenance of a master file for a population, or (b) *merging two files in order to extend the amount of information available for population units represented in both files*." (emphasis added)); Howard B. Newcombe and James M. Kennedy, "Record Linkage: Making Maximum Use of the Discriminating Power of Identifying Information," Comm. of the Assoc. for Computing Machinery 130:3381, https://dl.acm.org/doi/pdf/10.1145/368996.369026 at 563 (Oct. 16, 1959) ("Linkage of a *pair of records* relating to a particular individual or family involves two steps: first, a searching operation in which potentially linkable records are brought together for scrutiny, followed by a detailed comparison to decide whether the person or persons referred to on each are in fact the same." (emphasis added)); Halbert L. Dunn, "Record Linkage," Am. Journal of Public Health, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1624512/pdf/amjphnation00640-0051.pdf, at 1414 (Dec. 1946) (describing "record linkage" as linking certain statistics with "other facts about the same individuals").

[7] *See* https://pandas.pydata.org/about/.

one after the other without considering the presence of any common key between them.[8] The following figures illustrate the distinction between these two functions in practice:

Figure 2 - Python .join() Function[9]

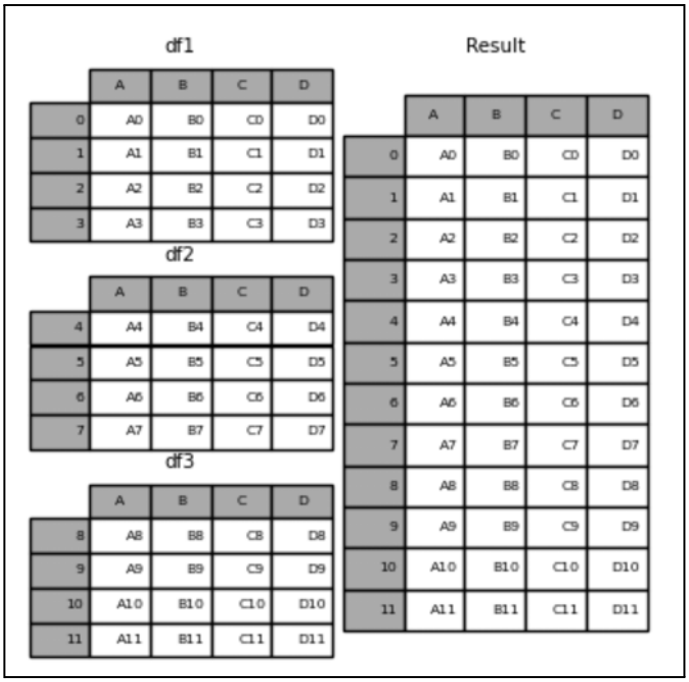


	A	B
K0	A0	B0
K1	A1	B1
K2	A2	B2

	C	D
K0	C0	D0
K2	C2	D2
K3	C3	D3

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3

Figure 3 - Python Default concat() Function



	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

48. As the figures above illustrate, "joining" records in Python causes input records that share a common key to be combined into a single record in the output table, and thus any "joined" output records include data from both input records. By contrast, Python's default

[8] The .join() function is using the more general merge() function. The concat() function concatenates along the rows axis by default. For more details about these functions see "Merge, join, concatenate and compare," https://pandas.pydata.org/docs/user_guide/merging.html.

[9] The rows of the "left" and "right" input DataFrames with common keys K0 and K2 are joined. (Shown join result with parameter how="outer".)

"concat" command merely adds records to the output table without generating any "joined" records that contain information from more than one input.

49. Similarly, in Structured Query Language ("SQL")—a programming language commonly employed for database management and analysis—the JOIN statement (also referred to as INNER JOIN) is used to combine data or rows from two or more tables based on a common field between them.[10]

50. I have examined a source code file called [redacted], which contains the instructions for sorting input logs data in the [redacted] log. Based on my review, it is my opinion that unauthenticated data is not joined with authenticated data in [redacted]

51. Below, I discuss in detail the key steps in adding inputs to generate this log, which support my opinion.

52. The first step is to call the function [redacted], which is used to extract from an authenticated log entry the corresponding GAIA ID, or, if the log entry is unauthenticated, the corresponding Zwieback ID. The full body of the function is reproduced here:

[redacted]

[10] *See generally* Mark Reed, *SQL: 3 books in 1 - The Ultimate Beginner, Intermediate & Expert Guides To Master SQL Programming Quickly with Practical Exercises*, Ch. 4 (2022); Abraham Silberschatz, Henry Korth, and Shashank Sudarshan. D*atabase system concepts*, Ch. 4 & 15 (2011).

53. An "if" statement controls whether or not a section of a program is performed, based on evaluation of a condition contained in the parentheses following the "if" command. If the condition is evaluated to be true, the section of the program contained in brackets following the parentheses will be performed. If the condition is evaluated to be false, that section of the program will not be performed. In the code snippet reproduced in paragraph 12, the first "if" statement (which I have reproduced in red font below) prescribes a logic where (i) if a GAIA ID is contained in a log entry, then (ii) the GAIA ID is stored in a variable called [redacted], and a variable called [redacted], which indicates whether this is an authenticated log entry, is set to true:

Because this is an "if" statement conditioned on presence of a GAIA ID, it will only run when a GAIA ID is present in a given log entry.

54. If the condition in a preceding "if" statement is false, an "else if" statement can be used to specify a new condition to control whether or not another section of the program is performed. If the condition for the original "if" statement is evaluated as true (and thus the

corresponding portion of the program is performed), any "else if" statement that follows that portion of the program will not be performed. In other words, the "else if" condition is only evaluated when the preceding "if" condition is false. In the code snippet reproduced in paragraph 12, the "else if" statement (which I have reproduced in blue font below) prescribes a logic where (i) if (and only if) a Zwieback ID (and not a GAIA ID) is contained in a log entry, then (ii) the Zwieback ID is stored in the [redacted] variable, and the [redacted] variable, which indicates whether this is an authenticated log entry, is set to false.

[redacted]

Because this is an "else if" statement, it will only run when the condition described in the preceding paragraph is not satisfied (*i.e.*, if there is ***no GAIA ID*** present in a given log entry), but a Zwieback ID is present in that log entry.

55. The second step in generating the [redacted] log is to call the function [redacted], which is used to (i) generate a new key which combines the [redacted] with the [redacted] for the log entry, and then (ii) assign the value of this key to the [redacted]. The code snippet below calls the function:

[redacted]

and the code snippet below assigns the value of the new key to the [redacted]:

[redacted]

A [redacted] is an identifier generated by Google for each log entry. Because, for all practical purposes, each log entry has a unique [redacted], this second step guarantees that it is practically impossible for [redacted] log entries to have the same [redacted]. For example, even if two log entries have the same GAIA ID, they will have a different [redacted] and because the value of the [redacted] is based on the combination of the [redacted] (in this example, the GAIA ID) and the [redacted], these [redacted] log entries will have [redacted] values. Similarly, even if two log entries have the same Zwieback ID, they will have a different [redacted] and, as a result, these two log entries will have different [redacted] values. Crucially, entries keyed to a Zwieback ID will never share a [redacted] with entries keyed to a GAIA ID, even in the absence of this second step that uses the [redacted], because any Zwieback ID is different than any GAIA ID.

56. The third and last step in generating the [redacted] log is to call the function [redacted], which is used to sort the log entries based on the [redacted] value, via the code snippet below:

[redacted]

The function [redacted] belongs to the well known MapReduce framework.[11] *See* GOOG-BRWN-00858535 (discussing MapReduce uses). Once sorted by the [redacted], the corresponding log entries are written to the log [redacted] one by one, as separate entries.

57. Based on my review of this code and GOOG-BRWN-00858535, it is evident that there is no joining of any log entries during the operation described above. The [redacted] log lists log entries from [redacted] authenticated and [redacted] unauthenticated logs one by one, as separate entries, based on the aforementioned sorting. No two

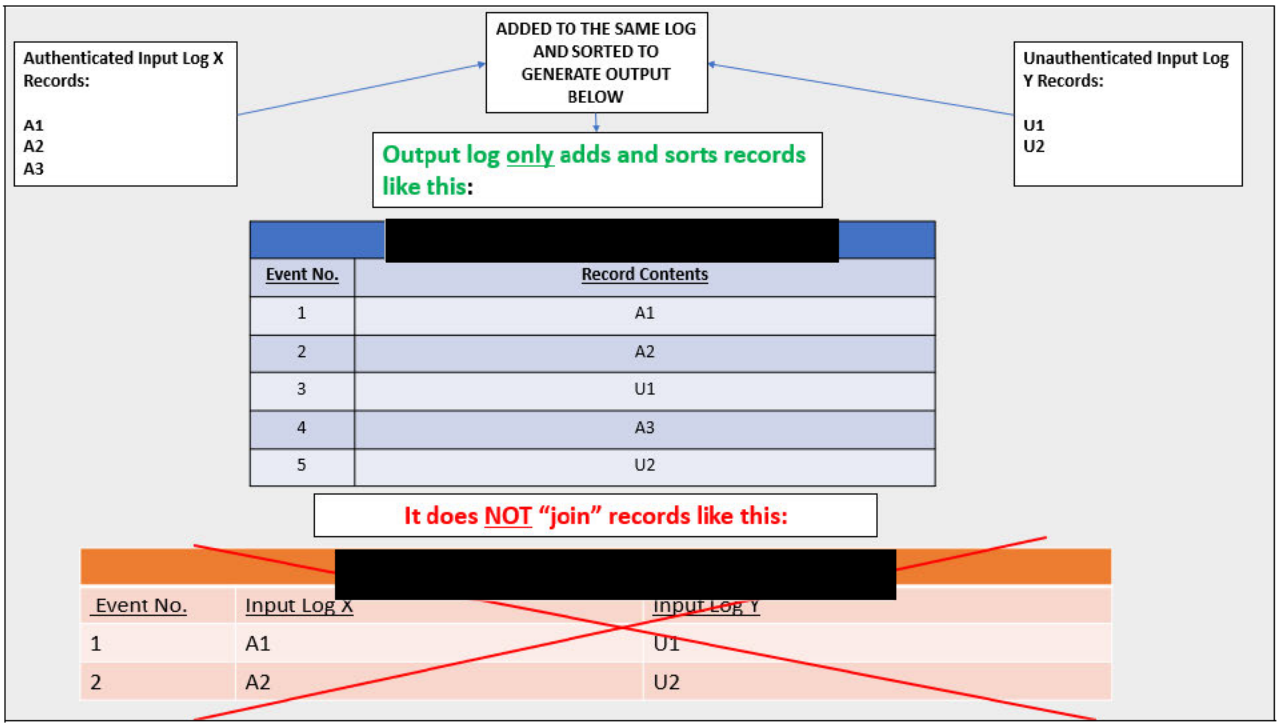
[11] *See* Jeffrey Dean and Sanjay Ghemawat, "MapReduce: simplified data processing on large clusters," OSDI '04: Sixth Symposium on Operating System Design and Implementation at 137-150 (2004).

log entries from the aforementioned logs are "joined" because records from the input logs are not combined together into a single record (*see* Figure 4, *infra*). In the extremely unlikely event that two log entries with the same GAIA ID also have the same [redacted], then these two authenticated entries could be joined. Similarly, in the extremely unlikely event that two log entries with the same Zwieback ID also have the same [redacted], then these two unauthenticated entries could be joined. However, there is no circumstance in which an authenticated log entry could be joined with an unauthenticated log entry, because the GAIA ID of the former will never match the Zwieback ID of the latter.

58. In summary, (i) it is extremely unlikely that any authenticated log entries will be joined together, (ii) it is extremely unlikely that any unauthenticated log entries will be joined together, and (iii) it is impossible for authenticated and unauthenticated log entries to be joined together. The code does not allow joining of authenticated data with unauthenticated data (or vice versa). My conclusion is based on the logic in the code itself, which is discussed above.

59. An illustration of how this works in practice for [redacted] is below:

Figure 4 - Adding and Sorting of Logs



60. My conclusions in paragraphs 57 and 58 above are consistent with the opinions stated in my Rebuttal Report regarding Google's policy and technical restrictions that prohibit joining of authenticated and unauthenticated data. *See, e.g.*, Dkt. 659-10 §§ III.A, III.C., III.F, III.G, IV.C. As explained above, the source code that defines the joining logic for [redacted] maintains a separation between records containing authenticated data and records containing unauthenticated data, and thus this data remains "segregated."

61. Certainly nothing in these logs or the source code would permit joining a given user's authenticated browsing activity with his or her unauthenticated browsing activity.

B. Sorting Records By Time Does Not "Link[ ] The Signed-Out Private Browsing Data With The User's GAIA ID."

62. Mr. Hochman also opines that "time ordered sequences of events from a user's signed-out private browsing and signed-in browsing (with the same IP address and user agent, as well as with many of the other parameters I discussed in my opening and rebuttal reports) may

appear in sequential order, linking the signed-out private browsing data with the user's GAIA ID." Hochman Second Supp. Rep. ¶ 45. For the reasons discussed below, I disagree with Mr. Hochman's opinion.

63. Based on my experience and training, and my understanding of the volume of queries received from Google's servers, the scenario that Mr. Hochman suggests "may" occur will never happen in practice. For time ordering to cause records from signed-in browsing and signed-out private browsing to "appear in sequential" order in the [redacted] log, there would have to be no records from other users added to this log during the time that elapsed between a signed-in user (i) switching to private browsing mode; and (ii) navigating to a website that uses Google web services (or vice versa). Millions of records are added to these logs every second, so there is almost no chance that [redacted] time sorting would ever cause these two records to appear in sequential order without any intervening records from another user.

64. Additionally, for the reasons discussed above, even if such sequential ordering were to occur (and it almost certainly would not), that would not constitute "linking" or "joining" as that term has been understood since the concept of record linkage was first explored in the 1940s.[12] Records that are ordered sequentially by time are still not "joined" because they are not combined into a single record via the use of a common key.

65. Nor would this time-sorting make "private browsing data . . . even more identifying" as Mr. Hochman claims. Hochman Second Supp. Rep. ¶ 45. As I explained in detail in my Rebuttal Report, signed-out private browsing data collected by Google web services is not keyed to a user's account or his/her identity. Mr. Hochman's proposed methodology for

[12] *See* Halbert L. Dunn, "Record Linkage," Am. Journal of Public Health, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1624512/pdf/amjphnation00640-0051.pdf, at 1414 (Dec. 1946) (describing "record linkage" as linking certain statistics with "other facts about the same individuals").

fingerprinting users via a combination of IP address and user agent (or other pseudonymous identifiers) would not work. *See* Rebuttal Report §§ III.A, III.G, III.I, III.J, IV.B, V.E–F. In my opinion, merely adding unauthenticated records and authenticated records to the same log does not resolve any of these problems (or make them any easier to solve). Mr. Hochman's claim that the [redacted] log somehow shows that the data that Google collects is even more identifying than he previously understood is therefore incorrect.

C. Storing Signed-In And Signed-Out Records In The [redacted] Log Does Not Contradict My Prior Opinions Regarding Google's Segregation Of Signed-In And Signed-Out Browsing Data

66. Mr. Hochman states that my prior opinion that "Google 'has taken steps to segregate signed-in from signed-out data'" is purportedly incorrect because "[t]he data is not 'segregated'" if it is "stored in the same exact log." Hochman Second Supp. Rep. ¶ 46 (citing Rebuttal Report ¶ 202). As discussed above, the [redacted] log does not permit any joining or linking of signed-in and signed-out browsing data. Moreover, there is no common key that would allow such joining or linking in any event because Mr. Hochman's assertion that an IP address and user agent combination could be used as a join key is incorrect. Additionally, the source code for this log shows that Google specifically designed the sorting function in this log to prevent joining of signed-in and signed-out browsing data, which demonstrates an additional step Google has taken to maintain the segregation referenced in my Rebuttal Report.

VI. THE ADDITIONAL [redacted] LOGS DO NOT CHANGE MY PRIOR OPINIONS

67. Mr. Hochman also asserts that "[t]he information that Google has provided about [redacted] additional logs with Incognito detection bits provides additional support for my already asserted opinions, including my opinions about Google's interception, storage, and use of private browsing data, Google's ability to identify private browsing traffic and class members, Google's

ability to delete the private browsing data it has already collected, and Google's ability to destroy algorithms developed through that data." Hochman Second Supp. Rep. ¶ 6. I have rebutted these opinions by Mr. Hochman at Sections III.A, III.C, III.F, III.G, III.H, III.I, and V.E-F.

68. Mr. Hochman also contends that the existence of one of these logs ("[redacted]") undermines my prior opinion that Google "has taken steps to segregate signed-in from signed-out data." *Id.* ¶ 46. I also disagree with Mr. Hochman's contention for the reasons discussed below.

69. The [redacted] additional logs containing the maybe_chrome_incognito bit and one additional log containing the is_chrome_non_incognito_mode bit identified by Mr. Hochman do not change any of the opinions stated in my Rebuttal Report submitted in this case. In particular, my Report includes several opinions regarding Plaintiffs' proposed method for identifying putative class members. *See* Report §§ III.A, C, F-H, J, and IV.C. For the reasons below, the existence of the additional [redacted] logs does not change any of the opinions I have offered in this case.

70. **Report Section III.A (Opinion 1): "Mr. Hochman's Opinion That Users Can Readily Be Identified From The Data At Issue (# 18) Is Incorrect."** The additional [redacted] logs do not change this opinion because the data-at-issue is orphaned and unidentified: (a) any unauthenticated data in these logs is still keyed to an unauthenticated identifier (or no identifier) for a signed-out user that is unique to the private browsing session; (b) these logs do not change the operation of the cookie jar and server-side processes for users in private browsing modes described in paragraphs 37 through 58 of my Report; and (c) in addition to the policy and technical restrictions described in Sections III.A, III.C., III.F, III.G, and IV.C of my Rebuttal Report, the source code discussed above provides another example of technical solutions Google has implemented to prevent the joining of authenticated and unauthenticated data.

71. **Report Section III.C (Opinion 3): "Mr. Hochman's Opinions On 'Private Browsing Profiles,' Server-Side Processes, And Data Joinability (# 10, 18, 19, 20) Are Inaccurate."** The additional [redacted] logs also do not change this opinion because (i) they do not show that Google maintains "cradle-to-grave" profiles of users that join signed-out private browsing mode activity with a Google account; and (ii) they do not change Google's server-side processes designed to prevent the joining of authenticated and unauthenticated data. Moreover, as discussed above, the lone "[redacted] log" included in these [redacted] logs that contains authenticated and unauthenticated records does not actually "join" any such records as that term has been consistently understood in the field for over 60 years, and the code that I have analyzed provides another example of technical restrictions that Google has implemented to prevent such joining.

72. **Report Section III.F (Opinion 6) "Mr. Hochman's Assertions On Fingerprinting Are Misleading And Unfounded."** The additional [redacted] logs also do not change this opinion because they do not show that Google engages in fingerprinting or undermine any of my opinions regarding the technical and policy constraints that Google implements to prevent the use of fingerprinting to re-identify users. Additionally, the source code analyzed above provides an additional example of technical restrictions that Google has implemented in line with its policies prohibiting fingerprinting and/or joining of authenticated and unauthenticated data.

73. **Rebuttal Report Section III.G, III.I (Opinions 7 and 9): "Mr. Hochman's Proposal to Identify Class I (Chrome Class) Is Unreasonable And Unreliable," and "Mr. Hochman's Proposal to Identify Class II (Non-Chrome Class) Is Unreasonable And Unreliable."** As explained further below, the additional [redacted] logs do not change this opinion because the same issues described in my Report will also affect any attempt to identify class members by applying the same fingerprinting methodology to these logs. *Id.* Specifically, Mr. Hochman's proposed fingerprinting methods would still rely on combinations of IPv4/IPv6

addresses and user agents that are "not sufficiently unique to identify class members because there are many situations where more than one user will have an identical IP address and user agent" (or other identifiers that are not suitable for the fingerprinting methodology that Mr. Hochman proposes for the reasons discussed in my Rebuttal Report) if it were applied to these [redacted] additional logs, and these [redacted] additional logs do not resolve this fundamental flaw because these flaws are inherent to the proposed use of a combination of an IP address and user agent (or other pseudonymous identifiers) as a join key to identify class members. *Id.* ¶ 111; *see also id* §§ V.E–F (appendices rebutting Mr. Hochman's claims regarding entropy and the ability to identify users via Mr. Hochman's proposed fingerprinting methodology).

74. **Report Section III.H (Opinion 8): "Mr. Hochman's Opinion That The 'maybe_chrome_incognito' Bit Reliably Detects Incognito Traffic (# 23) Is Incorrect."** These additional [redacted] logs also do not change this opinion because the maybe_chrome_incognito bit (and the is_chrome_non_incognito_mode bit) in these logs is still based on the absence of the X-Client-Data header. As I explained in my Report, this is not a reliable method for accurately identifying the use of Incognito mode because there are instances where the X-Client-Data header is not sent, but the user is not browsing in Incognito mode. These "false positives" render the maybe_chrome_incognito and is_chrome_non_incognito_mode bits unreliable for the reasons stated in my Report, and the existence of additional logs that contain the same bits does not change this opinion. *Id.* ¶¶ 142-145.

75. The existence of additional logs containing the maybe_chrome_incognito bit does not change my opinion that "[Mr.] Hochman's opinion that the 'maybe_chrome_incognito' bit reliably detects Incognito traffic is incorrect" because my opinion is based on the fundamental way that the maybe_chrome_incognito bit is computed. *Id.* As I stated in my Rebuttal Report, the maybe_chrome_incognito bit is a "boolean field that relies on the absence of the X-Client-Data

header to approximate and monitor traffic Google receives from Chrome instances in Incognito mode," and "the absence of the X-Client-Data header cannot be used to reliably detect Incognito traffic because there are a variety of cases in which the X-Client-Data header *is not* sent by a Chrome browser when a user is using a browser in non-Incognito mode (false positives)." *Id.* ¶ 143. As such, the maybe_chrome_incognito bit in these additional [redacted] logs also "cannot be used to reliably detect Incognito traffic because there are a variety of cases in which the X-Client-Data header *is not* sent by a Chrome browser when a user is using a browser in non-Incognito mode (false positives)." *Id.* (emphasis in original). In other words, these additional [redacted] logs also "can not be used to reliably detect Incognito traffic, let alone identify purported members of Class I" because they are affected by the same false positive problem as the previously-identified logs. *Id.* ¶ 145. And for these additional [redacted] logs, there is also "no way to exclude the false positives . . . because the reason for false positives is 'not observable from a server perspective.'" *Id.* (quoting Berntson June 16, 2021 Tr. 384:23-24).

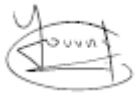
76. Similarly, the [redacted] log identified by Mr. Hochman does not change my opinions regarding the is_chrome_non_incognito_mode bit stated in my Rebuttal Report. *See, e.g.*, Rebuttal Report § III.H. As I previously explained, the is_chrome_non_incognito_mode bit also relies on the absence of the X-Client-Data header to approximate Incognito traffic. *Id.* ¶ 143 & n. 172. As such, it suffers from the same flaws described above.

77. **Report Section III.J (Opinion 10): "Mr. Hochman's Proposed Methods For Identifying Class Members (# 22) Do Not—And Cannot—Account For Shared Devices Or Accounts."** The additional [redacted] logs do not change this opinion because the same issues described in my Rebuttal Report will also affect any attempt to identify class members by applying the same fingerprinting methodology to these logs. *Id.* Specifically, Plaintiffs' proposed methodology still

does not account for shared devices, which render their methodology unreliable in light of widespread device-sharing and resulting collisions of IP address and user agent combinations. *See Id.* ¶¶ 162-180.

I declare under penalty of perjury that the foregoing is true and correct.

Executed on the 30th day of August 2023 at Los Angeles, CA.

By: 

Konstantinos Psounis, PhD